WHAT IS CLAIMED IS:

1. A method for determining a server computer which carried out a process most recently, applicable to a high availability computer system comprising a master server computer and a slave server computer each having a storage unit, the method comprising the steps of:

executing state-transition of said two server computers when a fault occurs in the server computer or the server computer is restored from the fault;

storing a priority determined by the state-transition into the storage unit;

determining, when the server computer is restored from the fault, whether or not the priority of the server computer restored from the fault is higher; and

determining that the server computer restored from the fault becomes a server computer to take over a process, when the priority of the server computer is higher.

2. The method according to claim 1, wherein the step of determining the priority further including the step of:

comparing the priorities of the server computers when the server computers are restored from the faults so as to determine which priority is higher.

3. The method according to claim 2, wherein the step of determining the priority further including the

steps of:

determining, when faults occur in the server computers and after that, one thereof is restored from the fault, whether or not the priority of the server computer restored from the fault is the highest priority; and

determining that the priority of the server computer is higher only when the priority is determined to be the highest priority.

4. The method according to claim 3, wherein each of the server computers constructed to assume four states, which are master state in which the server computer carries out the process and has a mate which takes over the process, single master state in which the server computer carries out the process and has no mate which takes over the process, slave state in which the server computer does not carry out the process but has information necessary for taking over of the process, and halt state in which the server computer does not carry out the process and holds no further information necessary for taking over of the process.

5. The method according to claim 4, wherein the step of storing the priority further including the steps of:

changing the priority of the server computer so as to indicate the highest priority when the state of the server computer is changed to the single master state;

changing the priority of the server computer so as to indicate the second highest priority when the state of the server computer is changed to the master state;

changing the priority of the sever computer so as to indicate the lowest priority when the state of the server computer is changed to the slave state; and

prohibiting a change of the priority of the server computer when the state of the server computer is changed to the halt state.

6. The method according to claim 4, wherein the step of executing the state-transition further including the steps of:

executing, when faults occurs in said server computers and after that, said server computers are restored from the faults, the first state-transition such that one of said server computers is changed from the halt state to the single master state while the other one is changed from the halt state to the slave state, based on at least the result of determining that the given server computer becomes a server computer to take over a process; and

executing, after the first state-transition is completed, the second state-transition such that the one server computer is changed to the master state while the other one remains in the slave state.

7. The method according to claim 4, wherein the

step of executing the state-transition further including the steps of:

executing the third state-transition such that the server restored from the fault is changed from the halt state to the single master state or keeping the current state, based on at least the result of determining that the server computer restored from the faults becomes a sever computer to take over a process, when faults occur in said server computers and after that, any one thereof is restored from the fault; and

executing, when after the step of executing the third state-transition or keeping the current state is completed, the other server computer of said server computers is also restored from the fault, either the fourth state-transition such that the other server computer is changed from the halt state to the slave state or the fifth state-transition in which any one of said server computers is changed to the single master state while the other one is changed to the slave state.

8. The method according to claim 3, wherein the step of executing the state-transition further including the step of:

executing, after faults occurs in said server computers and then, only any server computer is restored from the fault, the state-transition of the server computer such that the server computer is a server computer which continues the process, when the server computer is not capable of being determined to have a priority higher than the other server computer, because the priority stored in the storage unit of the server computer restored from the fault is not the highest priority and then, a forced start instruction for forcing the server computer to continue the process is given from outside.

9. A high availability computer system comprising a master server computer and a slave server computer each having a storage unit, each of said server computers comprising:

state-transition means for executing a state-transition of said server computers when a fault occurs in the server computer or the server computer is restored from the fault;

state writing means for storing a priority determined by the state-transition into the storage unit;

first determining means for determining, when the each server computer is restored from the fault, whether or not the priority of the each server computer is higher; and

second determining means for determining that the each server computer restored from the fault becomes a server computer to take over a process.

10. A high availability computer system according to claim 9, wherein the first determining means

compares the priorities of the server computers, when the server computers are restored from the faults, so as to determine which priority is higher.

11. A high availability computer system according to claim 10, wherein the first determining means, when faults occurs in the server computers and then only the each server computer is restored from the fault, determines whether or not the priority of the each server computer is the highest priority and only when the priority is the highest priority, determines that the priority of the each server computer is higher.

12. A high availability computer system according to claim 9, wherein each of the server computers constructed to assume four states, which are master state in which the server computer carries out the process and has a mate which takes over the process, single master state in which the server computer carries out the process and has no mate which takes over the process, slave state in which the server computer does not carry out the process but has information necessary for taking over of the process, and halt state in which the server computer does not carry out the process and holds no further information necessary for taking over of the process.

13. A high availability computer system according to claim 12, wherein the state-transition means, when faults occur in the server computers and then the

server computers are restored from the faults, executes such state-transition that one of the server computers is changed from the halt state to the single master state while the other one is changed from the halt state to the slave state, based on at least the determination result of the second determining means, and then executes such state-transition that the one server computer is changed to the master state while the other one remains in the slave state.

14. A high availability computer system according to claim 12, wherein the state-transition means, when faults occur in said server computers and then only the each server computer is restored from the fault, either executes such state-transition that the each server computer is changed from the halt state to the single master state or keeps the current state, based on at least the determination result of the second determining means, and then when the other server computer of the server computers is also restored from the fault, executes either state-transition that the other server computer is changed from the halt state to the slave state or state-transition in which any one of the server computers is changed to the single master state while the other one is changed to the slave state.

15. A computer readable recording medium storing a program for determining a server computer which carried out a process most recently, applicable to high

availability computer system comprising a master server computer and a slave server computer each having a storage unit, the program comprising:

code means for when a fault occurs in the server computer or the server computer is restored from the fault, making the server computer execute state-transition of said server computers;

code means for making the server computer store a priority determined by the state-transition into the storage unit;

code means for when the server computer is restored from the fault, making the server computer restored from the fault determine whether or not the priority of the server computer restored from the fault is higher; and

code means for making the server computer restored from the fault determine that the server computer restored from the fault becomes a server computer to take over a process, when the priority is higher.